

Remarks

Claims 1-3, 6-11, 14-15 and 17 remain in this application. Claims 4, 5, 12, 13, and 16 are hereby canceled without prejudice. Claims 1-3, 9, 10, 11, and 17 are hereby amended. No new matter is being added.

Claim Rejections--Section 103

The pending claims were rejected under 35 U.S.C. § 103 as being unpatentable over Carini in view of Chang. This rejection is respectfully traversed with respect to the claims as now amended.

Amended claim 1 now recites as follows.

1. A method of cross-file inlining during compilation of a program, wherein which files to open and close is determined based on affinity weightings between the files, wherein the affinity weightings depend on a number of potential inlines between the files, the method comprising: **performing an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph; and continuing the iterative process until the affinity weightings of all edges in the inline affinity graph go to zero.**

(Emphasis added.)

The claim limitations of "**performing an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph; and continuing the iterative process until the affinity weightings of all edges in the inline affinity graph go to zero**" are

discussed, for example, on page 43, line 8, through page 44, line 24 of the specification, which is reproduced below for convenience of reference.

... The top-level algorithm is a simple iterative process consisting of the following broad steps:

- An edge from the inline affinity graph is chosen. **One embodiment of the invention simply chooses the edge with the highest weight since the weight signifies the number of inlines that can potentially be satisfied with the source files corresponding to the chosen edge.** One potential pitfall is the possibility of this edge having dependences on other edges of the inline affinity graph that have not yet been considered. However, in the general case, the percentage of inlines (for a certain edge in the inline affinity graph) that have dependences are assumed to be approximately uniform throughout the inline affinity graph. Assuming that the dependence relationship is not extremely biased towards a few ag-edges, this scheme should work well. In scenarios where the bias is extremely high towards a few ag-edges, this scheme will still work correctly but with reduced efficiency.
- **The source files corresponding to the chosen edge are retrieved.** It may be noted that these two source files may be the same signifying that the inlines are intra-file i.e. the caller and the callee reside in the same file.
- **Before inline transformation can proceed, the source files corresponding to the inline need to be opened.** This is referred to as the preparation phase for the impending inline.
- The preparation phase returns a status which can be one of the following values: prepn_true (i.e. preparation was successful and the inline can proceed), prepn_false (i.e. preparation was not successful), prepn_out_of_memory (i.e. preparation was not successful because a potential out_of_memory situation could be reached), and prepn_cold (preparation was not completed since the source files are cold).
- If the status returned by the preparation phase is prepn_true, it means that the associated source files could be opened and the inlines can proceed. If the status is prepn_out_of_memory, it means that a hold state has been reached. Some files need to be closed before this inline can proceed. In this case, StatusHandler is called. If the status returned by the preparation phase is prepn_cold, it means that the associated source files are considered cold and hence will not be opened at this point. In this scenario, the next edge in the inline affinity graph is examined.
- StatusHandler is called when inlines cannot proceed in the current situation. StatusHandler usually closes some files and opens others in order to allow inlines to proceed. This routine is described in more detail later.
- **After every inline, the affinity graph and the dependence graph are updated. The inline affinity graph is updated by reducing the weight of the corresponding edge.** The dependence graph is implicitly updated. This means that the dependence test from edge x to edge y always checks first whether y is already inlined. If yes, there is effectively no dependence from edge x to edge y.
- **The iterative process continues while there are remaining inlines to be performed. When the weights of all the edges in the inline affinity graph reach zero, all inlines have been done.**

(Emphasis added.)

Applicant respectfully submits that these claim limitations are not taught in either Carini, or Chang, or the combination thereof.

Regarding Carini, applicant agrees with the Examiner's assertion in the first office action that Carini fails to disclose an inline affinity graph.

Regarding Chang, applicant respectfully submits that Chang fails to disclose the claim limitations of "**performing an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph; and continuing the iterative process until the affinity weightings of all edges in the inline affinity graph go to zero**".

Therefore, neither Carini, nor Chang, nor the combination thereof teaches the claimed invention per amended claim 1. Hence, applicant respectfully submits that amended claim 1 now overcomes this rejection.

Similarly, amended claim 2 now recites that the inline analysis includes "**performance of an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph, and continuation of the iterative process until the affinity weightings of all edges in the inline affinity graph go to zero**." Therefore, for similar reasons as discussed above in relation to claim 1, claim 2 also now overcomes this rejection.

Claim 9 is an apparatus claim which is amended similarly to method claim 1. In particular, claim 9 now recites processor-executable instructions comprising "**instructions to perform an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph; and instructions to continue the iterative process until the**

affinity weightings of all edges in the inline affinity graph go to zero.”

(Emphasis added.) Therefore, for similar reasons as discussed above in relation to claim 1, claim 9 is also now patentably distinguished over the cited art.

Claim 10 is an apparatus claim which is amended similarly to method claim 2. In particular, claim 10 now recites that the analyzer module is further configured “**to perform an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph, and to continue the iterative process until the affinity weightings of all edges in the inline affinity graph go to zero.**” (Emphasis added.) Therefore, for similar reasons as discussed above in relation to claim 2, claim 10 is also now patentably distinguished over the cited art.

Claim 17 is amended similarly as claim 10. In particular, claim 17 now recites an inline analyzer “**performing an iterative process including (i) choosing an edge in an inline affinity graph with a highest affinity weighting, (ii) retrieving and opening source files corresponding to the chosen edge, and (iii) updating the inline affinity graph, and which continues the iterative process until the affinity weightings of all edges in the inline affinity graph go to zero.**”

(Emphasis added.) Therefore, for similar reasons as discussed above in relation to claim 10, claim 17 is also now patentably distinguished over the cited art.

Claims 3 and 6-8 depend from claim 2, and claims 11 and 14-15 depend from claim 10. Therefore, for at least the reasons as discussed above in relation to independent claims claim 2 and 10, dependent claims 3, 6-8, 11 and 14-15 are also patentably distinguished over the cited art. In addition, further reasons that some of the dependent claims are patentably distinguished over the cited art are given below.

Regarding claims 3 and 11, applicant respectfully submits that the “compilation model” disclosed in Carini is not an “inline affinity graph” as claimed. In

particular, claim 3 requires that “edges in the inline affinity graph correspond to a **number of potential inlines** across corresponding files.” (Emphasis added.) In contrast, the “routine cost function” of Carini characterizes “the **suitability** of each procedure for inlining”, and the ““call site cost function” characterizes “the **suitability** of each call site for inlining”. (See page 6 of the latest office action, emphasis added.) Therefore, applicant respectfully submits that Carini does not teach or disclose the limitation of claim 3. For similar reasons, applicant respectfully submits that Carini does not teach or disclose the limitation of claim 11.

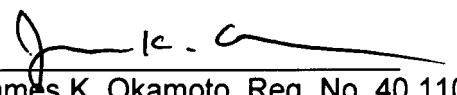
Conclusion

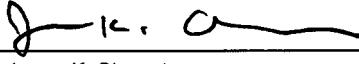
For the above-discussed reasons, applicant respectfully submits that the pending claims are now patentably distinguished over the cited art. Favorable action is respectfully requested.

The Examiner is also invited to call the below-referenced attorney to discuss this case.

Respectfully Submitted,

Dated: August 27, 2007


James K. Okamoto, Reg. No. 40,110
Tel: (408) 436-2111
Fax: (408) 436-2114

CERTIFICATE OF MAILING			
I hereby certify that this correspondence, including the enclosures identified herein, is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below. If the Express Mail Mailing Number is filled in below, then this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service pursuant to 37 CFR 1.10.			
Signature:			
Typed or Printed Name:	James K. Okamoto	Dated:	August 27, 2007
Express Mail Mailing Number (optional):			